CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Previously Presented) A data processing system for executing a program having branch instructions therein, each branch instruction specifying a target address in said program defining an instruction that is to be executed if that branch instruction causes said program to branch, said data processing system comprising:

a plurality of processing sections, each processing section comprising:

a local memory for storing instruction sequences from said program that are to be executed by that processing section, said instruction sequences comprising instructions of different lengths;

a function unit for executing instructions stored in said local memory; and

a pointer containing a value defining the next instruction in said local memory to be executed by said function unit, wherein each processing section executes part of said program;

each function unit executes instructions according to machine cycles, each function unit executing one instruction per machine cycle; and

said pointers in each of said processing sections are reset to a new value determined by said target address of one of said branch instructions when a function unit branches in response to that branch instruction.

2. (Previously Presented) The data processing system of claim 1 further comprising a memory for determining said new value of said pointers, said memory storing a mapping for each target address in said program specifying one of said pointer values for each of said pointers corresponding to that target address.

3. (Previously Presented) The data processing system of claim 1 wherein said program is divided into super instructions, each super instruction comprising a linear block of code that can only be entered at a starting address corresponding to said block of code and each block of code having one or more branch instructions, wherein said target address of at least one of said branch instructions corresponds to a starting address of a super instruction in said program.

4. (Previously Presented) The data processing system of claim 1 wherein at least one of said instruction sequences comprises at least one no op instruction.

5. (Original) The data processing system of claim 1 wherein said local memory of one of said processing sections comprises a cache memory.

6. (Previously Presented) The data processing system of claim 3 further comprising at least one processing section that remains idle for the duration of a super instruction.

7. (Previously Presented) The data processing system of claim 3 wherein a super instruction comprises a tuple for each processing section that is not idle for the duration of said super instruction and wherein each tuple identifies a function unit on which execution occurs and a number of memory words needed to represent corresponding operations to be executed on said function unit.

8. (Previously Presented) The data processing system of claim 3 wherein a super instruction indicates a total number of machine cycles for said super instruction.

9. (Previously Presented) The data processing system of claim 3 wherein said local memories are loaded at a start of said program.

10. (Previously Presented) The data processing system of claim 3 wherein said local memories are loaded at the time of a branch to a super instruction.

11. (Previously Presented) The data processing system of claim 3 wherein a fall-through super instruction is executed when a super instruction executes without branching.

12. (Currently Amended) A data processing system for executing a super instruction according to machine cycles, said super instruction comprising a linear block of code including instruction sequences to be executed by each of a plurality of processing sections and one or more branch instructions, ~~one instruction for each machine cycle,~~ the data processing system comprising:

a plurality of processing sections, each processing section comprising:

a local memory for storing instruction sequences of the super instruction that are to be executed by that processing section;

a function unit for executing instructions stored in said local memory according to machine cycles, each function unit executing one instruction per machine cycle; and

a pointer containing a value defining the next instruction in said local memory to be executed by said function unit, the pointers in each of said processing sections being reset to a new value determined by a target address of one of said branch instructions when a function unit branches in response to that branch instruction.

13. (Currently Amended) The data processing system of claim 12 wherein said linear block of code of said super instruction can only be entered at a starting address~~ and includes one or more branch instructions~~.

14. (Cancelled)

15. (Currently Amended) The data processing system of claim 12 [[13]] wherein said target address of at least one of said branch instructions corresponds to a starting address of another super instruction~~ in said program~~.

16. (Previously Presented) The data processing system of claim 14 further comprising a memory for determining said new value of said pointers, said memory storing a mapping for each target address specifying one of said pointer values for each of said pointers corresponding to that target address.

17. (Previously Presented) The data processing system of claim 12 wherein at least one of said instruction sequences comprises at least one no op instruction.

18. (Previously Presented) The data processing system of claim 12 said instruction sequences comprise instructions of different lengths.

19. (Previously Presented) The data processing system of claim 12 further comprising at least one processing section that remains idle for the duration of the super instruction.

20. (Previously Presented) The data processing system of claim 12 wherein said super instruction comprises a tuple for each processing section that is not idle for the duration of said super instruction and wherein each tuple identifies a function unit on which execution occurs and a number of memory words needed to represent corresponding operations to be executed on said function unit.

21. (Previously Presented) The data processing system of claim 12 wherein said super instruction indicates a total number of machine cycles for said super instruction.

22. (Previously Presented) The data processing system of claim 12 wherein said local memories are loaded at a start of a program that includes said super instruction.

23. (Previously Presented) The data processing system of claim 12 wherein said local memories are loaded at the time of a branch to said super instruction.

24. (Previously Presented) The data processing system of claim 12 wherein a fall-through super instruction is executed when said super instruction executes without branching.